

In the Claims

The claims have been amended as follows:

- 1 1. (Previously presented) A method for updating existing code in a computer
- 2 program after inputting new code which defines changes to said existing code
- 3 comprising the steps of:
 - 4 generating a target file list of target files to update;
 - 5 generating a dependency file list of files dependent on said target files;
 - 6 reading said dependency file list of files into a control file, wherein selected lines of
 - 7 said files are split into target strings having programming language substitutions
 - 8 and being appended to a requisition list and into prerequisite strings being stored
 - 9 in corresponding requisite arrays; and
 - 10 executing an algorithm where said algorithm matches said target files with said
 - 11 substituted target string in said requisition list in said control file, and then said
 - 12 algorithm updating those matched target files if it is determined that the
 - 13 corresponding prerequisite strings stored in the corresponding requisite arrays in
 - 14 said control file have been updated more recently than said substituted target
 - 15 string.

1 2. (Previously presented) The method of claim 1 further including source code and
2 object code, said target files being source code and said dependency files being object
3 code, said source code being selectively compiled to update said object code.

1 3. (Previously presented) The method of claim 1 further comprising the step of:
2 updating said prerequisite strings with new files defined by new code.

1 4. (Previously presented) The method of claim 2 wherein said algorithm utilizes a
2 search technique including pattern type variables which use generic rules to specify said
3 object code for updating.

1 5. (Previously presented) A method for generating changes and updating existing
2 files and code in a computer program, comprising the steps of:
3 reading existing target files and existing dependency files in said computer program;
4 reading a plurality of said dependency files associated with said target files into a
5 single control file, wherein selected lines of said dependency files are split into
6 target strings and prerequisite strings;
7 executing a utility program which updates said target files and said dependency files
8 associated with said target files, said utility program including an interpreted
9 scripting language specifying particular characters to search for in said target files
10 and said associated dependency files;

11 generating a requisition list of target strings having interpreted scripting language
12 substitutions and corresponding requisite arrays for said prerequisite strings using
13 said utility program; and
14 updating said target files by employing a search technique defined in said utility
15 program, said search technique includes specified target patterns such that said
16 specified target patterns identify said existing target files being updated, said
17 existing target files being updated if it is determined from said specified target
18 patterns that said prerequisite strings in said control file have been updated more
19 recently than said substituted target string.

1 6. Previously presented) The method of claim 5 wherein said specified target
2 patterns of said search technique includes pattern type variables which use generic rules
3 to specify said target files for updating.

1 7. (Previously presented) The method of claim 5 wherein said search technique
2 includes matching specified characters in said target files to said requisition list of target
3 strings such that said specified characters identify said existing target files being
4 updated.

1 8. (Cancel.)

1 9. (Previously presented) The method of claim 5 wherein said utility program
2 defines new target files to be added to said existing target files.

1 10. (Previously presented) The method of claim 5 wherein the utility program
2 prioritizes said target files to update while employing said search technique.

1 11. (Previously presented) The method of claim 5 wherein said utility program
2 includes a process procedure for an operator to call, said process procedure recursively
3 invokes said utility program and arguments.

1 12. (Previously presented) The method of claim 5 wherein said utility program is in
2 a scripting language selected from the group consisting of *updt*, perl, and Tcl.

1 13. (Previously presented) The method of claim 5 wherein said utility program
2 provides that said existing target files with a specific character are not considered to be a
3 file, and thereby are bypassed for any changes by the utility program.

1 14. (Previously presented) A computer program product for updating existing code
2 in a computer program after inputting new code which defines changes to said existing
3 code, said computer program product having:
4 computer readable program code means for generating a target file list of target files
5 to update;
6 computer readable program code means for generating a dependency file list of files
7 dependent on said target files;

8 computer readable program code means for reading said dependency file list of files
9 into a control file, wherein selected lines of said files are split into target strings
10 having programming language substitutions and being appended to a requisition
11 list and into prerequisite strings being stored in corresponding requisite arrays;
12 and

13 computer readable program code means for executing an algorithm where said
14 algorithm matches said target files with said substituted target string in said
15 requisition list in said control file, and then said algorithm updating said target
16 files if it is determined that the corresponding prerequisite strings stored in the
17 corresponding requisite arrays in said control file have been updated more
18 recently than said substituted target string.

1 15. (Previously presented) A program storage device readable by a machine,
2 tangibly embodying a program of instructions executable by the machine to perform
3 method steps for updating existing code in a computer program after inputting new code
4 which defines changes to said existing code, said method steps comprising:
5 generating a target file list of target files to update;
6 generating a dependency file list of files dependent on said target files;
7 reading said dependency file list of files into a control file, wherein selected lines of
8 said files are split into target strings having programming language substitutions
9 and being appended to a requisition list and into prerequisite strings being stored
10 in corresponding requisite arrays; and

11 executing an algorithm where said algorithm matches said target files with said
12 substituted target string in said requisition list in said control file, and then said
13 algorithm updating said target files if it is determined that the corresponding
14 prerequisite strings stored in the corresponding requisite arrays in said control file
15 have been updated more recently than said substituted target string.

1 16. (Previously presented) A method for updating target files in a computer
2 comprising:
3 generating a target file list of target files to update;
4 reading into a control file a list of files dependent on said target files;
5 splitting selected lines of said dependent files into target strings and prerequisite
6 strings;
7 performing programming language substitutions in said target strings;
8 appending said substituted target strings to a requisition list;
9 storing said prerequisite strings in corresponding requisite arrays;
10 executing an algorithm to match selected target files from said target file list to said
11 substituted target string in said requisition list;
12 retrieving said prerequisite strings from said corresponding requisite arrays;
13 updating said prerequisite strings by performing all possible programming language
14 substitutions to said prerequisite strings using said algorithm;
15 identifying those prerequisite strings that have been updated more recently than said
16 substituted target string to generate update rules using said algorithm; and
17 updating said target files from said target file list using said update rules.

1 17. (Previously presented) The method of claim 16 wherein after said list of files
2 dependent on said target files are read into said control file, said remaining subsequent
3 steps utilize said control file.

1 18. (Previously presented) The method of claim 16 further including updating said
2 target file list with new target files, the new target files being defined by said update
3 rules

1 19. (Previously presented) The method of claim 16 wherein said update rules
2 comprise target patterns to specify entire classes of dependencies.

1 20. (Previously presented) The method of claim 16 wherein the update rules are
2 specified using a scripting language selected from the group consisting of *updtk*, perl, and
3 Tcl.

1 21. (Previously presented) The method of claim 20 wherein said algorithm is
2 executed in said scripting language.

1 22. (Previously presented) The method of claim 21 wherein said rules have access to
2 said interpreted programming language to recursively invoke said algorithm on a new
3 target.

1 | 23. (Cancel)

1 24. (Previously presented) The method of claim 16 wherein said prerequisite uses
2 dynamic directory switching to specify multiple files in multiple directories.

1 25. (Previously presented) The method of claim 16 further including a directory, said
2 algorithm considering said directory to be out-of-date regardless of its time stamp such
3 that any rule associated with directory target is always triggered.

4 26. (New) A method for updating target files in a computer comprising:
5 generating a target file list of target files to update;
6 reading into a control file a list of files dependent on said target files;
7 splitting selected lines of said dependent files into target strings and prerequisite
8 strings;
9 performing programming language substitutions in said target strings;
10 appending said substituted target strings to a requisition list;
11 storing said prerequisite strings in corresponding requisite arrays;
12 executing an algorithm to match selected target files from said target file list to said
13 substituted target string in said requisition list;
14 retrieving said prerequisite strings from said corresponding requisite arrays;
15 updating said prerequisite strings by performing all possible programming language
16 substitutions to said prerequisite strings using said algorithm;

17 identifying those prerequisite strings that have been updated more recently than said
18 substituted target string to generate update rules using said algorithm, said update
19 rules support multi-directory builds from a single control file; and
20 updating said target files from said target file list using said update rules.